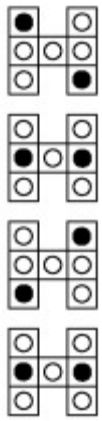
Übungsaufgabe 1: Hin-und-her

Diese Aufgabe benutzt unsere Anwendungsplatine "Würfel mit Morsecodeausgabe" und macht Euch mit den elementaren Assemblerbefehlen für die Ausgabe und Eingabe an den Port-Anschlüssen des ATtiny13 bekannt. Wie bei fast allen Anwendungen gibt es viele Möglichkeiten die gestellte Aufgabe zu lösen. Einige werden wir uns genauer ansehen. Aber zunächst sollt Ihr selbst probieren!



Im Bild links sind 4 Zustände gezeigt, die der Reihe nach angezeigt werden sollen, wobei es von einem Zustand zum nächsten entweder nach einer Wartezeit von etwa 0,5 sec. von selbst oder aber nach einem Tastendruck weiter gehen soll Wenn der letzte Zustand erreicht ist, soll es beim ersten wieder los gehen.

Die diagonalen LEDs leuchten, wenn PORTB.2 (PB2) bzw. PORTB.4 (PB4) auf 0 (low) geschaltet werden und diese Ports als Ausgang definiert sind. Ausschalten kann man die LEDs also indem man eine 1 in das jeweilige Bit im Ausgaberegister schreibt. Oder man deklariert den Port-Anschluss wieder als Eingang, wobei später der Ausgang nicht wieder erneut auf 0 geschaltet werden muss, denn das steht ja noch in dem Ausgangsregister so drin.

Die beiden LEDs, die zur SECHS gehören liegen am PORTB.3 (PB3) und werden genau so angesteuert wie die diagonalen LEDs. Allerdings funktioniert hier nicht die Lösung, eine 1 an den Ausgang PORTB.3 zu schicken, denn dann leuchtet die mittlere LED!

Da die Wartezeit bzw. das Abfragen der Taste vier Mal vorkommt, bietet sich dafür jeweils ein Unterprogramm an.

Zuerst solltet Ihr die automatische Weiterschaltung programmieren weil das am einfachsten ist. Eine Wartezeitroutine von 25 Millisekunden Dauer (0,025 Sek.) steht in dem Dokument **Anleitung zur Struktur eines Programmes** (assemblerprogrammierung.pdf auf der Homepage).

Beim Abfragen der Taste müsst Ihr das Prellen jeweils abwarten, wenn eine Änderung erkannt wurde - und zwar in jeder Richtung. Eine Dauer von 25 ms (Millisekunden) reicht meistens aus. Und das sind 30000 Takte, weil wir mit 1,2 MHz Taktfrequenz arbeiten und ein Takt also
1 : 1.200.000 = 0,000.000.833 Sekunden (0,833 Mikrosekunden) dauert.
Und 0,000.000.833 x 30.000 = 0,025 Sekunden.

Die Befehle, die ihr brauchen könnt sind:

ldi out sbi cbi sbis sbic rjmp rcall ret

sowie für die Wartezeitroutine (Beispiel siehe **Anleitung zur Struktur eines Programmes**):

ldi dec brne ret

Bitte beachtet die **Anleitung zur Struktur eines Programmes**

Wer beide Möglichkeiten in ein Programm packen will, könnte mal versuchen die Umschaltung auf "Automatik" durch langes Drücken der Taste auszulösen. Das ist aber schon eine große Herausforderung!

Viel Erfolg - oder für Funkamateure: vy 55 - wünscht

Hartwig, DH2MIC